LISTING OF CLAIMS

1. (Currently Amended) A method performed by an information handling system for processing a sequence of instructions that includes first and second instructions, wherein each of the first and second instructions is processable in a sequence of stages that includes first and second execution stages, and wherein the first instruction's second execution stage is processable substantially concurrent with processing the second instruction's first execution stage, comprising:

executing the first instruction during both of its first and second execution stages, in which a first at least one of an arithmetic operation, a logical operation, an address calculation operation, a masking operation, and a shifting operation of the first instruction is performed in response to first source operand information, and in which first destination operand information is output in response thereto; and

executing the second instruction during a selected one of its first and second execution stages, in which a second at least one of an arithmetic operation, a logical operation, an address calculation operation, a masking operation, and a shifting operation of the second instruction is performed in response to second source operand information, and in which second destination operand information is output in response thereto, so that the second instruction is executed: during only its first execution stage in response to the second source operand information being independent of the first destination operand information; and during only its second execution stage in response to the second source operand information being dependent on the first destination operand information; and wherein the second destination operand information is written for storage in a memory after the second instruction's second execution stage, even if the second instruction is executed during its first execution stage.

2. (Previously Presented) The method of Claim 1, wherein executing the second instruction comprises:

executing the second instruction during the selected one of its first and second execution stages, in response to an encoding of the second instruction.

3. (Previously Presented) The method of Claim 1, wherein the memory is a cache.

.

4. (Canceled).

5. (Canceled).

6. (Previously Presented) The method of Claim 1, wherein executing the second instruction comprises:

executing the second instruction during only its second execution stage in response to the second source operand information being dependent on the first destination operand information, but only if the system includes a suitable resource for executing such instruction during its second execution stage.

7. (Previously Presented) The method of Claim 1, wherein the sequence of stages includes multiple execution stages, including at least the first and second execution stages and an additional execution stage.

8. (Previously Presented) The method of Claim 7, wherein the additional execution stage precedes the first execution stage.

9. (Previously presented) The method of Claim 7, wherein the additional execution stage follows the second execution stage.

10. (Previously Presented) The method of Claim 7, wherein the additional execution stage is a first additional execution stage, and wherein the first additional execution stage precedes the first execution stage, and wherein a second additional execution stage follows the second execution stage.

11. (Canceled).

12. (Previously Presented) The method of Claim 1, wherein the second instruction is executable in a single machine cycle of the system, and wherein the first instruction is executable in only multiple machine cycles of the system.

13. (Previously Presented) The method of Claim 1, wherein the sequence of stages is processed in one machine cycle of the system per stage.

14. (Previously Presented) The method of Claim 1, wherein the sequence of stages is the same for the first and second instructions.

15. (Currently Amended) A method performed by an information handling system in assembling a sequence of instructions that includes first and second instructions, wherein each of the first and second instructions is processable in a sequence of stages that includes first and second execution stages, and wherein the first instruction's second execution stage is processable substantially concurrent with processing the second instruction's first execution stage, comprising:

assembling the first instruction for execution during both of its first and second execution stages, in which a first at least one of an arithmetic operation, a logical operation, an address calculation operation, a masking operation, and a shifting operation of the first instruction is to be performed in response to first source operand information, and in which first destination operand information is to be output in response thereto; and

assembling the second instruction for execution during a selected one of its first and second execution stages, in which a second at least one of an arithmetic operation, a logical operation, an address calculation operation, a masking operation, and a shifting operation of the second instruction is to be performed in response to second source operand information, and in which second destination operand information is to be output in response thereto, so that the second instruction is to be executed: during only its first execution stage in response to the second source operand information being independent of the first destination operand information; and during only its second execution stage in response to the second source operand information being dependent on the first destination operand information; and wherein the second destination operand information is to be written for storage in a memory after the second instruction's second execution stage, even if the second instruction is executed during its first execution stage.

16. (Previously Presented) The method of Claim 15, wherein assembling the second instruction comprises:

assembling the second instruction during the selected one of its first and second execution stages, in response to an encoding of the second instruction.

17. (Previously Presented) The method of Claim 15, wherein the memory is a cache.

18. (Canceled).

19. (Canceled).

20. (Previously Presented) The method of Claim 15, wherein assembling the second instruction comprises:
assembling the second instruction for execution during only its second execution stage in response to the second source operand information being dependent on the first destination operand information, but only if the system is specified as including a suitable resource for executing such instruction during its second execution stage.

21. (Previously Presented) The method of Claim 15, wherein the sequence of stages includes multiple execution stages, including at least the first and second execution stages and an additional execution stage.

22. (Previously Presented) The method of Claim 21, wherein the additional execution stage precedes the first execution stage.

23. (Previously Presented) The method of Claim 21, wherein the additional execution stage follows the second execution stage.

24. (Previously Presented) The method of Claim 21, wherein the additional execution stage is a first additional execution stage, and wherein the first additional execution stage precedes the first execution stage, and wherein a second additional execution stage follows the second execution stage.

25. (Canceled).

26. (Previously Presented) The method of Claim 15, wherein the second instruction is executable in a single machine cycle of the system, and wherein the first instruction is executable in only multiple machine cycles of the system.

27. (Previously Presented) The method of Claim 15, wherein the sequence of stages is processable in one machine cycle of the system per stage.

28. (Previously Presented) The method of Claim 15, wherein the sequence of stages is the same for the first and second instructions.

29. (Currently Amended) An information handling system for processing a sequence of instructions that includes first and second instructions, wherein each of the first and second instructions is processable in a sequence of stages that includes first and second execution stages, and wherein the first instruction's second execution stage is processable substantially concurrent with processing the second instruction's first execution stage, comprising:

first circuitry for executing the first instruction during both of its first and second execution stages, in which a first at least one of an arithmetic operation, a logical operation, an address calculation operation, a masking operation, and a shifting operation of the first instruction is performed in response to first source operand information, and in which first destination operand information is output in response thereto; and

second circuitry for executing the second instruction during a selected one of its first and second execution stages, in which a second at least one of an arithmetic operation, a logical operation, an address calculation operation, a masking operation, and a shifting operation of the second instruction is performed in response to second source operand information, and in which second destination operand information is output in response thereto, so that the second circuitry is for executing the second instruction: during only its first execution stage in response to the second source operand information being independent of the first destination operand information; and during only its second execution stage in response to the second source operand information being dependent on the first destination operand information; and wherein the second circuitry is for writing the second destination operand information for storage in a memory after the second instruction's second execution stage, even if the second instruction is executed during its first execution stage.

30. (Previously Presented) The system of Claim 29, wherein the second circuitry is for executing the second instruction during the selected one of its first and second execution stages, in response to an encoding of the second instruction.

31. (Previously Presented) The system of Claim 29, wherein the memory is a cache.

32. – 33. (Canceled).

34. (Previously Presented) The system of Claim 29, wherein the second circuitry is for executing the second instruction during only its second execution stage in response to the second source operand information being dependent on the first destination operand information, but only if the system includes a suitable resource for executing such instruction during its second execution stage.

35. (Previously Presented) The system of Claim 29, wherein the sequence of stages includes multiple execution stages, including at least the first and second execution stages and an additional execution stage.

36. (Previously Presented) The system of Claim 35, wherein the additional execution stage precedes the first execution stage.

37. (Previously Presented) The system of Claim 35, wherein the additional execution stage follows the second execution stage.

38. (Previously Presented) The system of Claim 35, wherein the additional execution stage is a first additional execution stage, and wherein the first additional execution stage precedes the first execution stage, and wherein a second additional execution stage follows the second execution stage.

39. (Canceled).

40. (Previously Presented) The system of Claim 29, wherein the second instruction is executable in a single machine cycle of the system, and wherein the first instruction is executable in only multiple machine cycles of the system.

41. (Previously Presented) The system of Claim 29, wherein the sequence of stages is processed in one machine cycle of the system per stage.

42. (Previously Presented) The system of Claim 29, wherein the sequence of stages is the same for the first and second instructions.

43. - 56. (Canceled).